

**Yahya FEKRANE**

E-mail : [fekyah0@gmail.com](mailto:fekyah0@gmail.com)

Tel N : 0707957177

# **RAPPORT**

## **De mini projet**

### **c++**

## **Objectif de mini projet**

---

*Application de la gestion des études*  
*« CRUD »*

---

Monsieur  
ABDALI

# TABLE DES MATIERES

INTRODUCTION.....	1
PARTIE DE CONCEPTION.....	2
PARTIE TECHNIQUE .....	3
CONCLUSION.....	4

# INTRODUCTION :

La réalisation d'un mini projet permet de maitrise les techniques étudiées ,voir des nouvelles technique .mon projet est base sur une problématique connue qui est :

## Gestion des étudiant

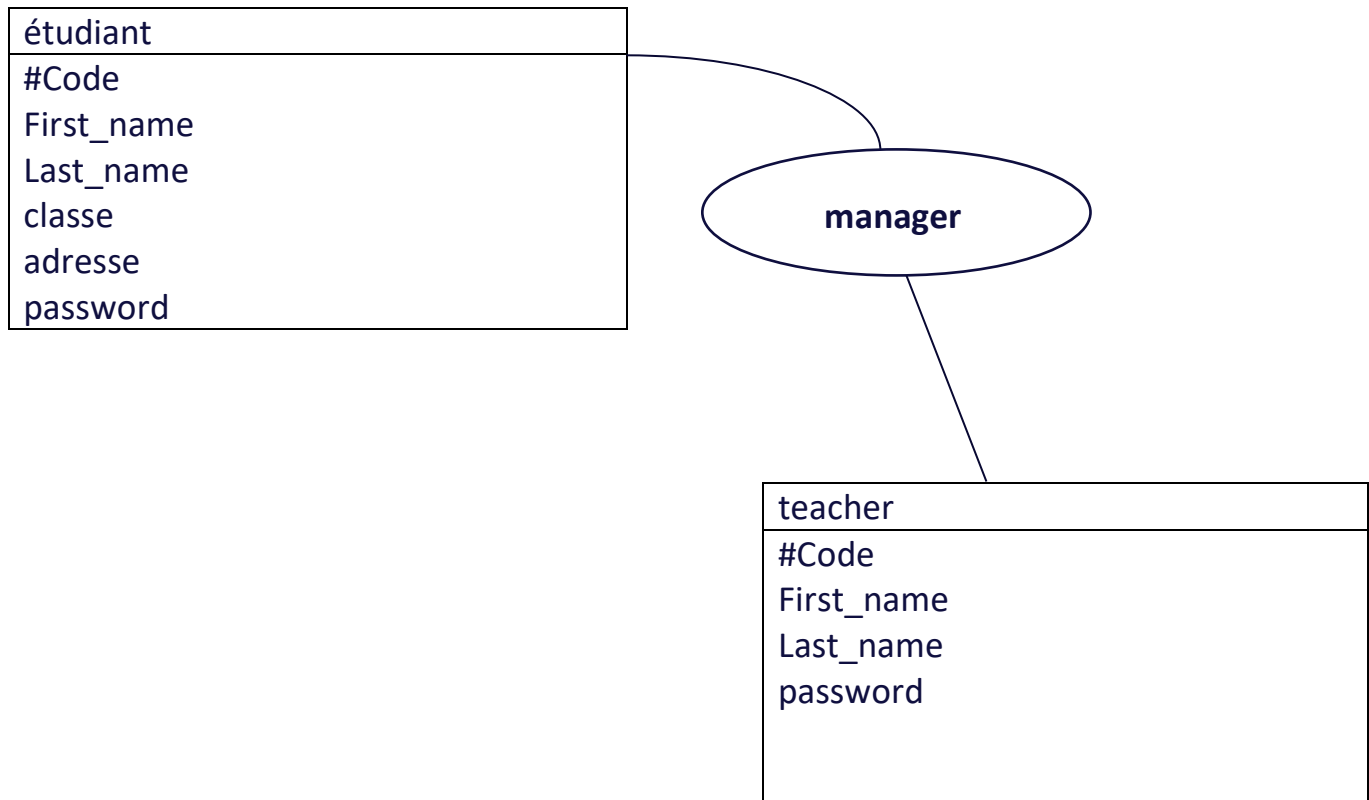
### La problématique

La gestion des données est un critère essentiel pour toute entreprise ou établissement il se peut que ça soit une gestion de stock, gestion des ressources humaines... Pour ce projet on a opté pour le cas d'un lycée qui aura besoin de gérer l'ensemble de ces étudiants à travers une interface simple et pratique. Cette gestion sera bien sûr traitée à travers une base de données stockée sur le local ou sur un serveur privé de l'établissement.

Le problème se posent au niveau de l'administration, surtout lorsque le nombre des étudiants, des professeurs et des matières augment et ça devient de plus en plus délicat de traité tous ces données manuellement, et stocké par la suite dans l'archive de l'école. C'est ici que l'informatisation des données devient un élément nécessaire, et le traitement automatique améliore énormément le rendement de l'établissement que ça soit au niveau du gain du temps ou au niveau des ressources humaines choisi pour effectuer les tâches.

Avant de se mettre à coder des lignes et des lignes de code, il faut tout d'abord limiter et tracer un cadre de développement qui donnera la vue générale du projet et les objectifs souhaité par ce dernier. C'est pour cela que la partie primordiale du projet est sa conception car une conception idéale donne automatiquement un projet réussi. Afin d'avoir une conception Optimale on se pose de différentes questions que notre projet doit absolument y répondre.

On veut faire une application qui faire la gestion des étudiants « C.R.U.D ». Création ou l'ajoute du nouveau étudiant , la lecture ou l'affichage de la liste des étudiants stockées , modification des information des étudiants, et la suppression. Donc il est claire qu'on a besoin d'une base de donnée qui contient les information de cet étudiant et qu'il va contenir les champ suivants (code, prénom, nom, classe , adresse , mot de passe) et un autre pour contenir les informations du professeur (code ,prénom ,nom ,mot de passe)



Pour ce projet on a essayé d'utiliser des technologies diverses de développement d'application professionnel, tout d'abord on travaillera avec la programmation Orienté Objet et comme environnement de développement on a choisi le langage C++. L'environnement de développement nécessite aussi une interface graphique qui sera présentée sur le FrontOffice, on a choisi pour la réalisation de cette interface le Framework de Nokia-Digia« le Qt »

#### Une petite présentation de Qt à partir de Wikipédia :

Qt est un Framework orienté objet et développé en C++ par QtDevelopmentFrameworks, filiale de Digia. Il offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Qt est par certains aspects un framework lorsqu'on l'utilise pour concevoir des interfaces graphiques ou que l'on architecture son application en utilisant les mécanismes des signaux et slots par exemple. Qt permet la portabilité des applications qui n'utilisent que ses composants par simple recompilation du code source. Les environnements supportés sont les Unix (dont Linux) qui utilisent le système graphique X Window System ou Wayland, Windows, Mac OS X et également Tizen. Le fait d'être une bibliothèque logicielle multiplate-forme attire un grand nombre de personnes qui ont donc l'occasion de diffuser leurs programmes sur les principaux OS existants. Qt est notamment connu pour être la bibliothèque sur laquelle repose l'environnement graphique KDE, l'un des environnements de bureau les plus utilisés dans le monde Linux. *Logicielle multiplate-forme attire un grand nombre de personnes qui ont donc l'occasion de diffuser leurs programmes sur les principaux OS existants.* Qt est notamment connu pour être la bibliothèque sur laquelle repose l'environnement **graphique KDE, l'un des environnements de bureau les plus utilisés dans le monde Linux.**





Pour le système de gestion de base de données (SGBD) on a utilisé SQLLITE .

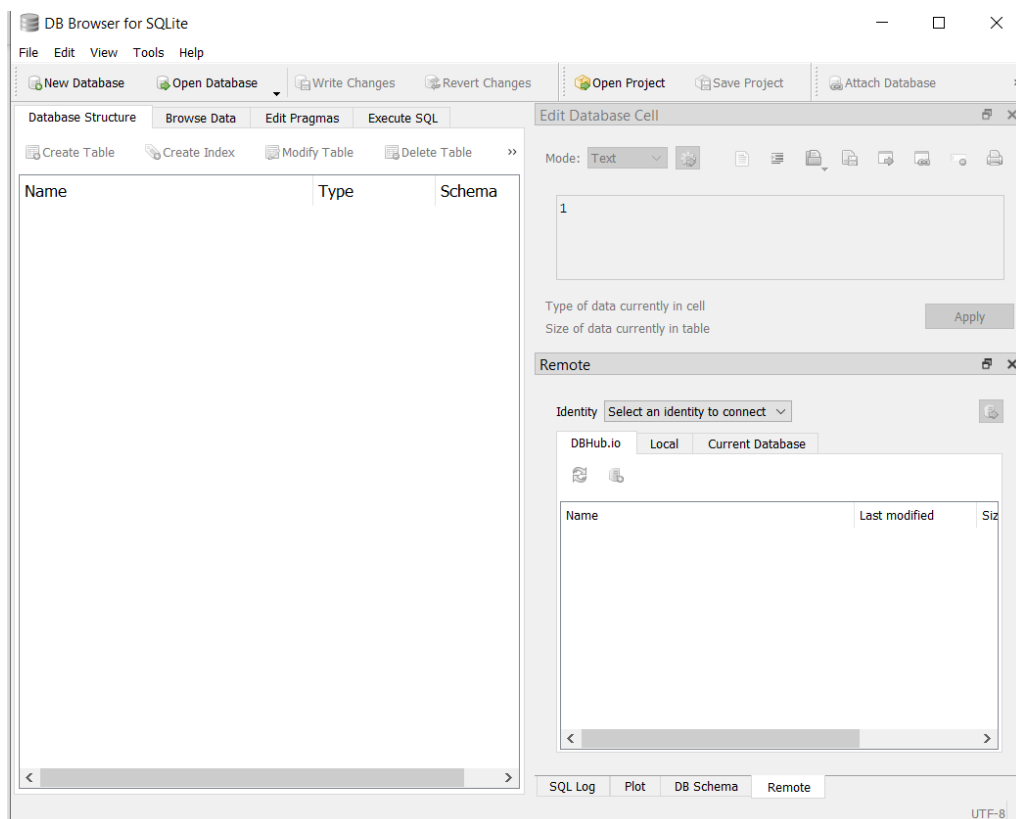
- La relation entre C++ et le SGBD

Ainsi, Qt offre une connexion rapide est simple à la base de donnée cela à travers les lignes de code suivantes :

```
mydb =QSqlDatabase::addDatabase("QSQLITE");
mydb.setDatabaseName("C:/Users/fekya/OneDrive/Bureau/studentManager/biblio.db");
```

## Application des règles de Gestion

DB Browser (sqlite) sera notre interface qui va nous permettre de créer les différentes tables de la base de données en respectant la structure qu'on a déjà posé lors de la réalisation du MCD.



Pour la création du backgrounds je choisir de travailler avec le logiciel adobe Photoshop 2022



### **La fenêtre sur Qt-Creator**

La réalisation des fenêtres sur Qt peut être faite par deux méthodes :

- A travers Qt-Designer
- A travers Qt-Creator

La différence entre eux c'est que Qt designer offre une interface de design graphique très simple à réaliser et à partir de laquelle on réalise plusieurs fenêtres de notre application.

Alors que sur Qt-Creator on aura besoin de tout codé ce qui nous offre une flexibilité et un contrôle totale de l'application.

Pour ce projet on opte à l'utilisation de Qt-Creator pour bénéficier du contrôle offert par ce dernier.

La réalisation d'une fenêtre sur Qt-Creator se fait de la manière suivante :

- On réalise le Layout général de l'application
- On définit les sous-Layouts

Projet c++ - Gestion des étudiants d'un lycée ENSA-Kenitra 2012-2013

12

- On remplit chaque sous-Layout par les buttons ou zone texte nécessaire

### **Envoie de requêtes**

La communication avec la base de données se fait à travers des requêtes SQL, Qt-creator à une méthode très simple pour gérer ces requêtes. Après avoir établie la connexion à la base de donnée on peut donc lancer des requêtes et cela à travers les deux fonctions hérité de la class QSqlQuery :

- Query.prepare()
- Query.exec()

Et on peut récupérer les résultats de sortie de la requête à l'aide de la fonction de la même class :

- Query.next()

### **Les requêtes du Projet**

D'après ce qu'on a vu précédemment, on aura besoin à effectuer quatre fonctions principales sur les différentes tables de la base de données.

- **Requête d'ajout d'un élément :**

La requête responsable de l'ajout d'un attribut dans la base de donnée est « INSERT INTO » l'interprétation de cette requête sur Qt se fait par la requête suivante :

```
INSERT INTO `table_en_question` (`valeur1`, `valeur2`) VALUES (:val1, :val2)
```

Puis on indique que `:val1` et `:val2` sont les variables données par l'utilisateur et cela se fait par la fonction « `bindValue()` » qui est une fonction hérité de la class des requêtes `QSqlQuery`.

- **Requête de modification d'élément :**

```
UPDATE `table_en_question`  
SET `attribut1`=:val1, `attribut2`=:val2  
WHERE id=:id_element_a_modifie
```

- **Requête de suppression d'un élément :**

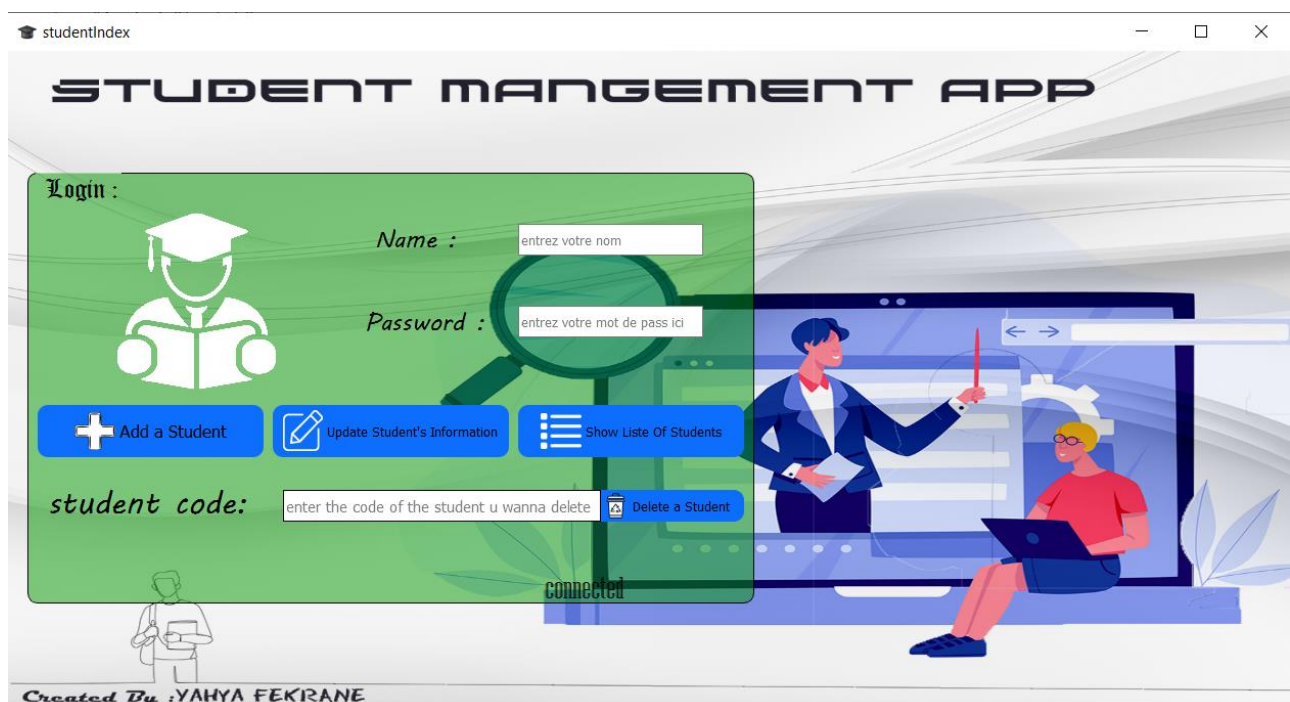
```
DELETE FROM `table_en_question` WHERE `id`=:id_element_a_supprimer
```

- **Requête d'affichage de l'ensemble des éléments :**

Pour l'affichage la requête est une « SELECT » mais le principe d'afficher tous les éléments sous forme d'un tableau qui est un peu spécial, car on aura à définir cela au niveau du code est non pas le SQL.

On sait que le SQL renvoi plusieurs valeurs sous forme d'un tableau, et on sait d'avance les dimensions du tableau qu'on souhaite avoir comme sortie, donc on aura besoin de définir ce tableau sur Qt avec la class `QTableWidget` puis on remplit en boucle chaque élément de ce tableau par les éléments renvoyés par la requête.

fenêtre Principale :



La fenêtre principal



L'objectif de cette fenêtre :

- une formulaire pour que l'étudiant peuvent connecter et faire les action crud. Je vue que les information pour cette connexion peut être :
  1. le prénom du étudiant .
  2. son mot de passe .
- des buttons pour accéder a les autre page :
  1. Une pour ajouter un nouveau étudiant .
  2. Une pour afficher la liste des étudiants sauvegarder.
  3. Une pour modifier les donnes d'un étudiant .
  4. Une pour supprimer un étudiant a travers son code . « cette Button doit être associer avec un champ pour écrire le code ».
- Des champs disent les états du page :
  1. Un pour afficher :
    - a) la connexion avec la base de donnes si il est connecter il afficher « connected ».
    - b) le mot de passe et le prénom si il sont correct ou pas.
    - c) L'existantes de deux étudiant avec les mêmes coordonnées .
  2. Une autre pour afficher un message durant la suppression du étudiant :
    - a) Un message au cas ou les champs de formulaire sont vide .
    - b) Un message de la suppression.
- Un messageBox au cas ou la suppression est s'effectuer.

### Explication du travaille de cette page :

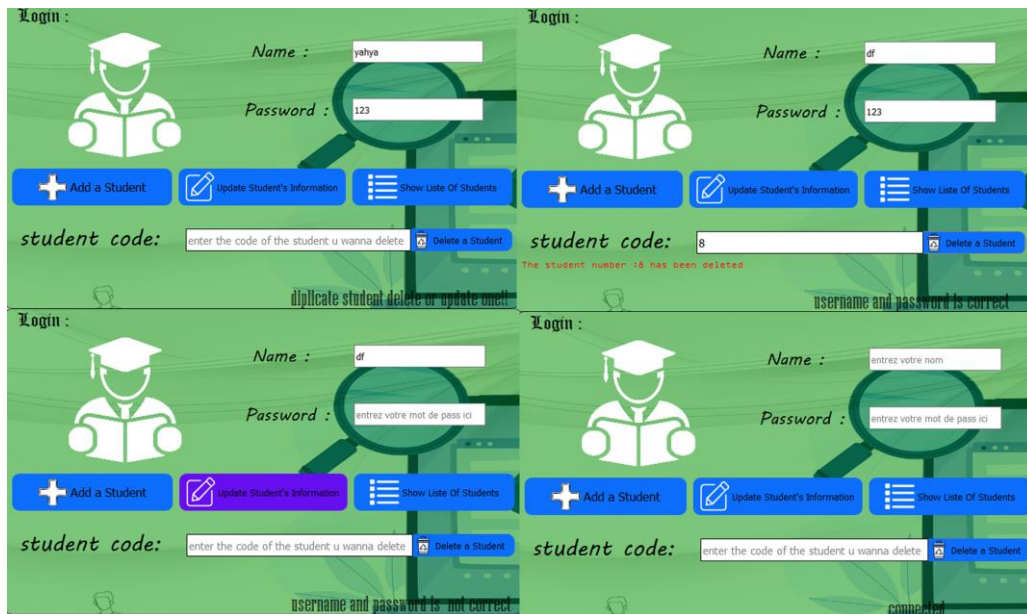
Avant tout j'effectuer la connexion de la fenêtre avec la base de donne si la connexion est effectuer au dessus du page on aura un message « **connected** » .

Puis les champs Name et Password doit être remplisser avec les donnes de l'étudiant qui veut accéder a les fonctions du projet .



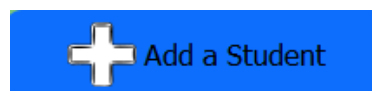
a la clique de toute les buttons le programme vérifier si les données entre est correct si il sont correcte le message au dessus change «**username and password are correct** » si il les données

sont incorrects le message devient « **username and password are not correct** » et si les on a la duplication du coordonnées on affiche ce message «**duplicate student delete or update one !!!** »

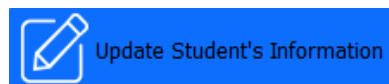


### Les buttons :

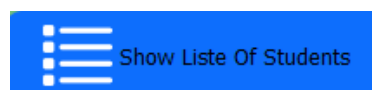
- Add a Student : permettre de ouvrir la fenêtre addstudent.



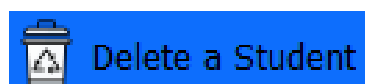
- Update Student's Information : permettre de ouvrir la fenêtre StudentUpdate.



- Show Liste of Students : permettre de ouvrir la fenêtre StudentListe.



- Delete a Student :



### Addstudent fenêtre :

add student

# ADD A NEW STUDENT

HOME

student information :

first name:  first name:

adresse:

filier :  anne :

password :

Created By : YAHYA FEKRANE

Contient une formulaire pour insertion du données de nouveau étudiant avec 3 buttons « submit » « close » « home » ;

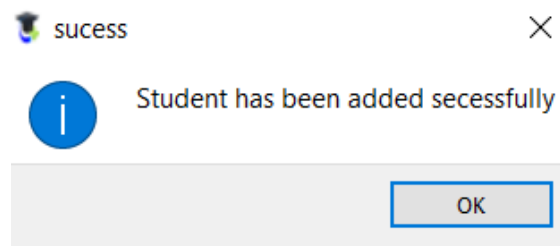
1. Submit : pour insertion des données saisis à la base de données avec la vérification si les champs first name, last name et password ne sont pas vides et si filière et année ne sont pas 'choose one ...'

```
if(fname==" " || lname==" " || p==" " || filier=="choose one ..." || year=="choose one ..."){
    ui->label->setText("fields are empty!!!");
}
```

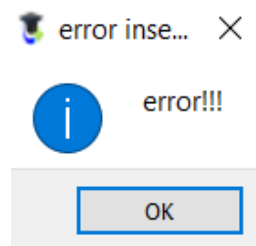
Au cas où les champs sont vides le programme affiche ce message au haut du message :



a) Si les données sont bien insérer le programme affiche un QMessageBox :



b) Si il y a un problème de l'insertion le programme affiche un QMessageBox :



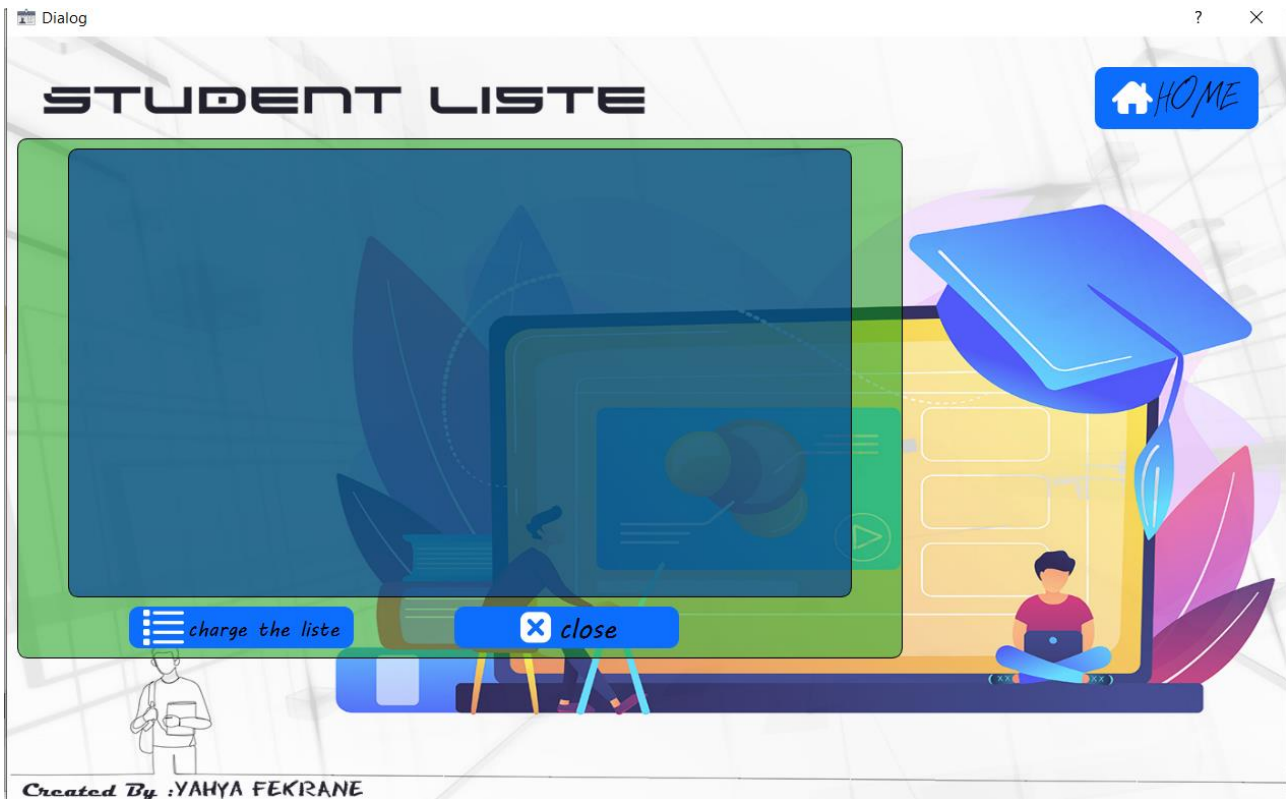
2. Close : pour quitter la fenêtre.



3. Home : pour revient a la page principale.



StudentUpdate fenêtre :



Cette fenêtre contient 3 button :

1. Charge the liste : permettre de remplir le tableView avec les données de la base de donnée.

	StudentNumber	First Name	Last Name	StudentClass	Adress	Password
1	1	omar	tijani	12	f3	123
2	3	yahya	fekrane	gg	iir3	123
3	4	amine	azillal	GÉNIE INDUSTRIEL fir...	y	ry
4	7	abdilah	tlaha		INGÉNIERIE AUTOMATISMES ET ...	123
5	9	ayoub	ayoub		INGÉNIERIE INFORMATIQUE ET ...	123

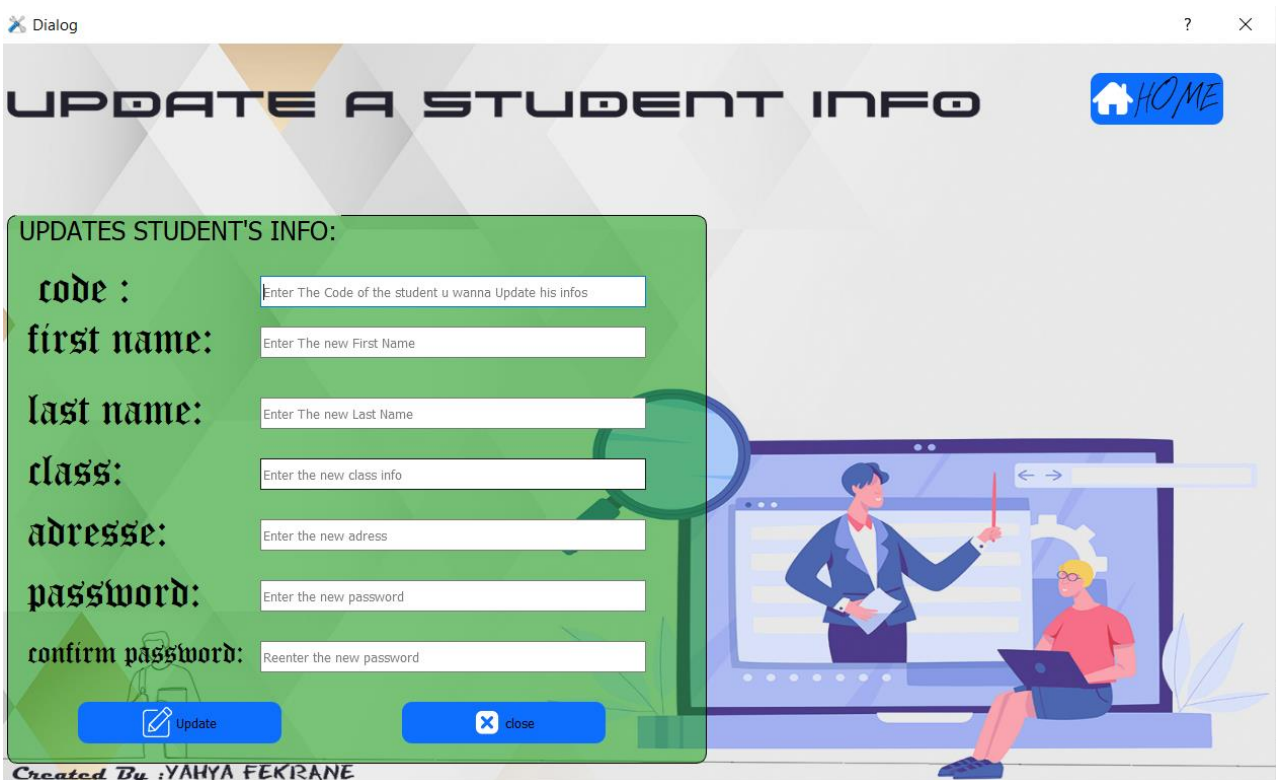
2. Close :pour fermer la fenetre .

 close

3. Home : pour revient a la page principale.



STudentUpdate fenêtre :



Dialog

## UPDATE A STUDENT INFO

UPDATES STUDENT'S INFO:

code :

first name:



last name:

class:

adresse:

password:

confirm password:

 Update  close

Created By : YAHYA FEKRANE

Cette fenêtre contient 3 Button :

1. Update : d'après le code de l'étudiant le programme change les autre informations .

```
QSqlQuery q;
q.prepare("update etudiant set first_name=:fname,last_name=:lastname,adresse=:adresse,class=:classe,password=:password"
        "where code=:code");
q.bindValue(":fname",fname);
q.bindValue(":lastname",lname);
q.bindValue(":classe",classe);
q.bindValue(":adresse",adresse);
q.bindValue(":code",code);
q.bindValue(":password",password);
```

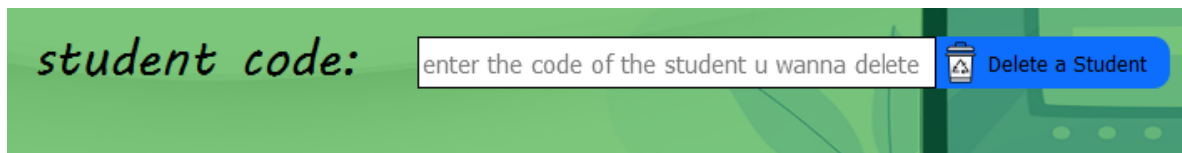
2. Close : pour fermer la fenetre .

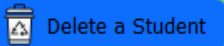


3. Home : pour revient a la page principale.

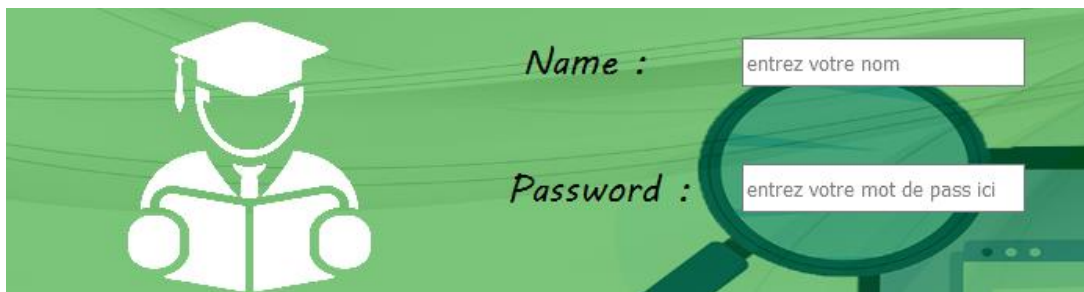


Fonction SudentDelete : « principale fenêtre »



*student code:*  

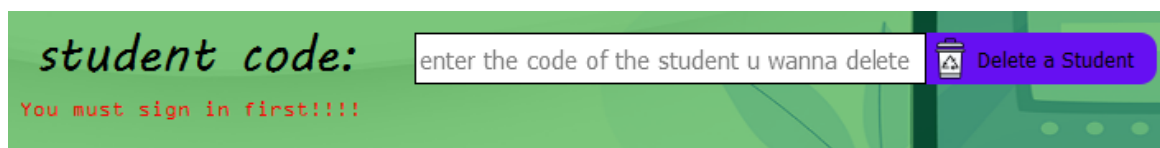
a la clique sur le Button la fonction vérifier premièrement si les coordonnées entrer dan la formulaire sont correcte :

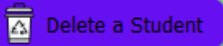


*Name :*

*Password :*


Si ils sont vide la fonction affiche cette message :

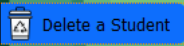


*student code:*  

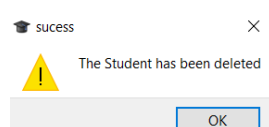
You must sign in first!!!!

Sinon la suppression effectue est la fonction affiche un message et un QMessageBox :



*student code:*  

The student number :8 has been deleted





# CONCLUSION

Le projet d'application de gestion d'étudiant, avait pour but de créer une application qui permet de gérer l'ensemble des étudiants .

Pour atteindre notre objectif on a opté pour une solution : base de donnée gérée par une application externe (Remote Control) , ce qui nous a permis d'appliqué et de développé nos connaissance du concept orienté objet ainsi que les langage SGBD et plus précisément le SOLLITE.

Ce projet nous a été bénéfique d'un point de vue conception et design, puisque c'est notre toute première application avec interface graphique, développant ainsi notre côté artistique et créatifs.

Lors de la réalisation, on a été confronté a plusieurs obstacles, surtout le problème d'intégration du SOLLITE avec le QT.

Bien que le travail effectuer est assez important, mais vu les contraintes du temps, il reste extensible.

Voila quelques idées proposées pour l'amélioration :

- Limiter l'accès au données au staff du établissement.
- Ajouter la gestion des niveaux et des classes.
- Ajouter une interface séparé pour les professeurs, permettant la saisi des notes.
- Ajouter l'option de calcule automatique des note générale pour chaque module et générale de chaque étudiant.
- Elargir notre application afin de gérer un établissement du niveau supérieur tel que notre école.